OCR Error Correction for Unconstrained Vietnamese Handwritten Text

Quoc-Dung Nguyen
Van Lang University
Ho Chi Minh, Vietnam
Technical University of Ostrava
Ostrava-Poruba, Czech Republic
dungnq.vtrd@gmail.com

Duc-Anh Le
Center for Open Data in the Humanities
Tokyo, Japan
leducanh841988@gmail.com

Ivan Zelinka
Technical University of Ostrava
Ostrava-Poruba, Czech Republic
ivan.zelinka@vsb.cz

ABSTRACT

Post-processing is an essential step in detecting and correcting errors in OCR-generated texts. In this paper, we present an automatic OCR post-processing model which comprises both error detection and error correction phases for OCR output texts of unconstrained Vietnamese handwriting. We propose a hybrid approach of generating and scoring correction candidates for both non-syllable and real-syllable errors based on the linguistic features as well as the error characteristics of OCR outputs. We evaluate our proposed model on a Vietnamese benchmark database at the line level. The experimental results show that our model achieves 4.17% of character error rate (CER) and 9.82% of word error rate (WER), which helps improve both CER and WER of an attention-based encoder-decoder approach by 0.5% and 3.5% respectively on the VNOnDB-Line dataset of the Vietnamese online handwritten text recognition competition (VOHTR2018). These results outperform those obtained by various recognition systems in the VOHTR2018 competition.

CCS CONCEPTS

• **Computing methodologies** → *Language resources*; *Neural networks*; • **Applied computing** → *Optical character recognition.*

KEYWORDS

Unconstrained Vietnamese handwriting, OCR, Post-processing, Error detection, Error correction

ACM Reference Format:
Quoc-Dung Nguyen, Duc-Anh Le, and Ivan Zelinka. 2019. OCR Error Correction for Unconstrained Vietnamese Handwritten Text. In *The Tenth International Symposium on Information and Communication Technology (SoICT 2019), December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3368926.3369686

1 INTRODUCTION

Optical Character Recognition (OCR) is the process of transforming typed, handwritten or printed text from scanned documents or images into digital text using various image processing and pattern recognition techniques [3, 5]. However, the OCR process often results in misspellings and linguistic errors in OCR-generated texts due to misrecognized characters, falsely identified text images as well as limitations of text recognition techniques. Post-processing is an important step to improve the quality of existing OCR output texts by detecting and cleaning the errors.

The increasing popularity of pen-based and touch-based devices has led to the crucial demand in processing so-called digital ink (a time sequence of pen/touch points) recently. The handwriting recognition systems have been developed with the ability to recognize, exchange and search for handwritten text from digital ink in order to meet various requirements and applications in education, entertainment, business and so on. Unconstrained handwritten text is written naturally in each writer's style without any restriction. Hence, the unconstrained handwritten text usually contains many variations in size, shape, slant, skew, and stroke order (see Fig. 1).

Unconstrained handwritten text recognition can be categorized into two main types: online and offline recognition. The first type of recognition makes use of spatial and temporal information of points of pen trajectory and strokes as input features to the recognition systems, for example, English [4], Chinese [20], Korean [8], etc. In the offline recognition type, only offline images (e.g. converted from online handwritten text) are available for image processing and recognition, such as English [2], Iranian [1], Indian [22], etc. In fact, the offline type of handwritten text recognition can be seen as a subtask of the OCR.

Vietnamese is a Latin script language. However, unlike other Latin script languages such as English, French or Spanish, Vietnamese contains a large amount of diacritic marks (DM), which are added to characters to transcribe all the sounds or to indicate variations in speech. They are placed over, under or through characters. For unconstrained handwritten text, the place and order of these DM strokes could be varied due to different writers or even different writing times of the same writer. In other words, DMs are often not positioned right where they should, and their sizes and shapes are also varied. They can be written after writing a few strokes of subsequent characters, or even after a sentence (called delayed DMs). These distorted and delayed DMs cause difficulties in recognizing unconstrained Vietnamese handwriting.

A few works have been proposed for Vietnamese online handwritten character recognition [17, 18, 21]. They came up with some solutions for solving the DM problem at character level. However, these approaches require pre-segmented handwritten text, they are not capable of dealing with cursively handwritten text in practice.

SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam

ACM ISBN 978-1-4503-7245-9/19/12…$15.00
https://doi.org/10.1145/3368926.3369686

Figure 1: Vietnamese words written in different styles.

Phuong et al. in [19] have proposed a SVM-based model for isolated offline Vietnamese handwritten character recognition. The authors employ three SVM classifiers in order to determine and recognize body characters and DM parts in handwritten character images. They achieve 90.51% of recognition rate on their private database.

The attention-based encoder-decoder (AED) model with DenseNet encoder is recently proposed by Le et al. [11] to overcome the pre-segmentation requirement of the previous studies. This AED model is an offline approach by recognizing unconstrained Vietnamese handwritten text images at word level and without using a language model. The handwritten images are converted from the corresponding online handwritten texts of the VNOnDB database organized by Nguyen et al. [13], and used as an evaluation benchmark in the Vietnamese online handwritten text recognition competition[1] (VOHTR2018) [14]. In addition, the converted text images help eliminate the delayed problem of DM strokes written in the online patterns. Nevertheless, the word error rate of the AED model is still higher than 10%, although this result on the VNOnDB-Line dataset is better than that of the other handwriting recognition systems such as Nguyen et al.'s, GoogleTask2 and IVTOVTask2 in the VOHTR2018 competition.

According to our best knowledge, there are no published researches on OCR post-processing specifically for unconstrained Vietnamese handwriting, or even only a few studies in the literature reported on OCR post-processing for Vietnamese OCR-generated texts [23]. In [23], the authors employ two correction schemes for non-syllable errors and real-syllable errors. The weighting-based correction scheme makes use of two linguistic features, syllable similarity and syllable frequency; while the contextual corrector employs the language modelling scheme based on perplexity score. In the second corrector, they implement Depth First Traversal strategy to examine all combinations of candidates. This leads to high computation cost (low speed) due to the explosion of combinations when the number of nodes (syllables) grows more than 10. They use their own dataset as well as the metrics (recall, precision and F1) for evaluating the performance of two schemes, so that it makes difficulty in comparison with other approaches.

In this paper, we propose an automatic OCR post-processing model including both error detection and error correction for OCR output texts of unconstrained Vietnamese handwriting. The proposed hybrid model is based on OCR error characteristics as well as linguistic features to generate and rank correction candidates of OCR errors. The model is employed and evaluated on OCR output texts at line level. We make use of the OCR output lines resulted from the AED model with DenseNet encoder called the baseline AED model as aforementioned. The experimental results show that our model improves the performance of the baseline AED model by 0.5% of CER and 3.5% of WER. Since our model does not rely on any specified parameters of the AED model (OCR process), it is not only applicable to OCR texts of unconstrained Vietnamese handwriting, but also to Vietnamese OCR texts in general.

The rest of the paper is structured as follows. Section 2 presents in detail our proposed model for OCR post-processing. Section 3 discusses the experimental results in comparison with other approaches. Finally, Section 4 gives our conclusion and discussion.

2 PROPOSED MODEL

In this section, we describe briefly the AED model and in detail the processing phases of our proposed OCR post-processing model, including tokenization, error detection, candidate generation and error correction (see Fig. 2). During the phases of error detection, candidate generation and error correction, we need to consult the word *n*-gram data. For our word-based model, we rely on the VietTreeBank[2] (VTB) corpus [15] to construct the dictionaries of word unigrams, bigrams and trigrams along with their occurrence frequency. In Vietnamese, each space-separated token (word unigram) is in monosyllabic form. Therefore, we call the word unigram dictionary as the syllable vocabulary from now on.

2.1 Attention-based Encoder-Decoder Model

The AED model in [11] consists of two components: Convolutional Neural Network-based (CNN) encoder and a Long Short-Term Memory-based (LSTM) decoder. The encoder is based on DenseNet for extracting invariant features from a handwritten text image. The LSTM-based decoder incorporated with an attention model generates output text. The DenseNet encoder has better performance in feature extraction than the CNN encoder and the row BLSTM encoder also proposed by Le et al. [10].

In the DenseNet, they employ a convolutional layer with 48 feature maps and a max pooling layer to process input image. Then three dense blocks are followed, each with growth rate (output feature map of each convolutional layer) K = 96 and the depth (number of convolutional layers in each dense block) D = 4. Transition layers, each containing convolutional and average pooling layers, are interleaved between the dense blocks. Direct connections among the dense blocks help the network reuse and learn features across layers.

[1] https://sites.google.com/view/icfhr2018-vohtr-vnondb/home

[2] Source: https://vlsp.hpda.vn/demo/?page=resources

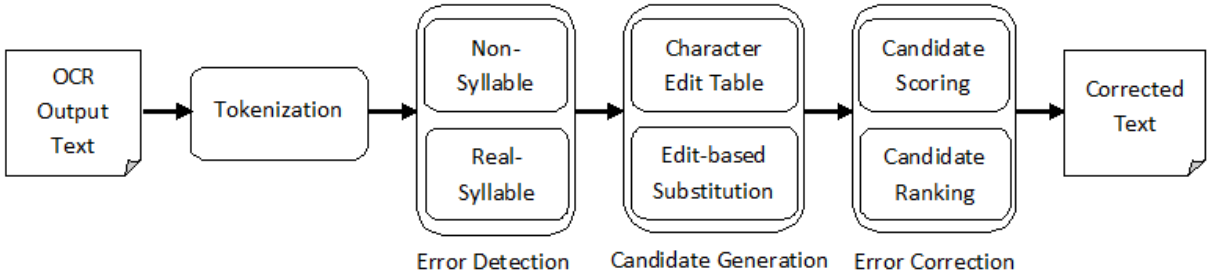

Figure 2: The processing phases in our proposed OCR post-processing model.

The attention-based LSTM decoder outputs one character at each time step t. The output character y_t is predicted based on the current output O_t and the context vector C_t, wherein C_t is a weighted sum of the sequence of outputs and their weights produced by an attention model.

2.2 Proposed OCR Post-processing Model

2.2.1 Tokenization. In the tokenization phase, the OCR text is tokenized on space with no restriction on punctuation. Except for the punctuation marks occurring at the end of token, including full-stop ".", comma ",", semi-colon ";", colon ":", double quote """, question "?", exclamation "!", closing parenthesis ")" and newline "\n", they will be removed before the error detection is processed. Each space-separated token is called a syllable.

2.2.2 Error Detection. From the OCR output texts resulted from the AED model, we observe the different kinds of syllable errors. These errors are caused by wrong edit operations of character insertion, deletion and substitution due to misrecognized characters and falsely identified text line images. They can be categorized into two main classes of errors, non-syllable errors and real-syllable errors. The non-syllable errors do not occur in a standard syllable dictionary (e.g. "còn" is wrongly recognized as "ccn" by the OCR engine); while the real-syllable errors refer to the syllables that exist in a standard syllable dictionary but are incorrect in the context meaning of text line (e.g. "sĩ" in the source text line "trách nhiệm của bác sĩ" is wrongly recognized as "sẽ" in the OCR text line). Table 1 gives the examples of non-syllable and real-syllable OCR errors. The purpose is to correct these errors as they impact on readable capability of the OCR output texts.

In the error detection phase, it will scan through the syllables in the OCR output texts and identify a syllable as an error if the syllable does not occur in the syllable vocabulary (non-syllable errors). For detecting real-syllable errors, given a syllable in the OCR text, we examine all the bigrams and trigrams of the syllable to see if they exist in the bigram and trigram dictionaries. If none of them does, the syllable is considered as erroneous. As detailed, the bigrams of the syllable s_i comprise two bigram contexts of s_i, "s_{i-1} s_i" and "s_i s_{i+1}", wherein s_{i-1} and s_{i+1} are the preceding and following syllables of s_i. The trigrams of s_i are in three contexts, "s_{i-2} s_{i-1} s_i", "s_{i-1} s_i s_{i+1}", and "s_i s_{i+1} s_{i+2}", wherein s_{i-2}, s_{i-1}, s_{i+1} and s_{i+2} are the syllables surrounding s_i.

There are other kinds of syllable errors formed by unexpected operators, such as:

- Incorrect upper-/lower- cases (e.g. "Sông" in the OCR text line "Có lắm cây dừa ở sâu cách xa Sông rạch").
- Splitted/merged syllable errors.
  For example, "côppha" is splitted into "công pha"; and "To nhưng" is merged into "Tonhưng".
- Deleted/inserted syllable errors.
  For instance, "còn" in the source text line "không chỉ làm nhà tình thương mà còn làm đường" is incorrectly removed in the OCR output line; and "tiết" is wrongly inserted in the OCR output line "Những khoản chi tiết cần thiết".

Table 1: Examples of non-syllable and real-syllable OCR errors.

OCR errors	Examples
Non-syllable	"để" recognized as "đẩo"
	"lo" recognized as "(0"
	"tránh" recognized as "trính"
Real-syllable	"Sau đó, được những người trong làng đi làm ăm" (OCR output line) "Sau đó, được những người trong làng đi làm ăn" (Source text line)
	"Những người hạn đồng hành đàng yêu. Người bạn già đồng hành đều tiên" (OCR output line) "Những người bạn đồng hành đáng yêu. Người bạn già đồng hành đầu tiên" (Source text line)

2.2.3 Candidate Generation. In this phase, we first construct a character edit table (CET) by applying a sliding window with sizes of one or two characters to the training texts. The training texts include the ground truths (GT) of input text line images and corresponding OCR outputs of the same input text lines. By aligning the GT text lines and the corresponding OCR output lines at the character level, we obtain the character edit patterns that transform the GT text lines into the OCR output lines in account with their observed frequency counts in the training data. Each character edit

consists of a GT pattern (one or two characters as well) from the GT text and corresponding error pattern (length of one or two characters) from the OCR text. The GT-error pattern pairs can contain the punctuation marks and even the space character. When correcting OCR errors, error patterns will be substituted with GT patterns called correction patterns. The CET table carries the error characteristics resulted from the OCR process of the baseline AED model.

In the close look at the training data, we encounter all types of character edit operations that consist of character insertion, deletion and substitution (see Table 2). The transposition operation can be considered as a special case of the substitution operation.

Table 2: Examples of the different types of character edits learned on the VNOnDB-Line dataset.

Operation	Edit Pattern	Example
Insertion	"i" insertion	"bà"→ "bài"
	"oàn" insertion	"Hoàng"→ "Hoàngoàn"
Deletion	"c" deletion	"chị"→ "hị"
	"~" DM deletion	"cũng"→ "cung"
	"ồng" deletion	"đồng"→ "đ"
Substitution	"á"→ "ú"	"Quá"→ "Quú"
	"d"→ "ch"	"dừng"→ "chừng"
	"k"→ "t" and "ở"→ "ợ"	"khởi"→ "thợi"

In the next step, we generate correction candidates for the detected OCR errors using the character edits obtained from the CET table. We make an erroneous token from each OCR error (syllable) before candidates are generated. For a non-syllable error, the erroneous token is also the non-syllable error. For a real-syllable error, it is combined with its preceding or following context syllables to create the erroneous token. Candidates of the erroneous token (called s_e) are generated by substituting character patterns scanned along the sequential characters of s_e with correction patterns, where the error patterns and the correction patterns are matched in pairs in the CET table. The candidate generation is controlled by the maximum number of edit operations (*edit_distance* parameter) applied to the erroneous token. If the generated candidate occurs in one of n-gram dictionaries, it is selected as the correction candidate; otherwise, it is discarded. All the selected correction candidates will be scored and ranked according to the OCR error characteristics and the linguistic features as discussed in the error correction section below.

2.2.4 Error Correction. We adopt and modify several important features that have been successfully employed for OCR text correction. These linguistic features capture the diverse characteristics of the language as explained in the related works [7, 12, 16]. We first choose the word level features composed of word similarity and context n-gram frequency. In order to adapt to the syllable vocabulary in our model for Vietnamese language, we call the word similarity feature as syllable frequency. Then we suggest the edit probability feature that is similar to the character confusion probability from the noisy channel model employed in [9, 16].

Our adopted features above are described as follows:

- Syllable Similarity:
  Longest Common Subsequence (LCS) measures the similarity of two strings. [6] introduced various modifications of Longest Common Subsequence (LCS) including Normalized Longest Common Subsequence (NLCS) and Normalized Maximal Consecutive Longest Common Subsequence (NMCLCS). The syllable similarity between correction candidate s_c and erroneous token s_e is calculated as a weighted sum of NLCS and variations of NMCLCS as proposed in [7]:

$$\begin{aligned} &Sim(s_c, s_e) = \\ &\alpha_1 * NLCS(s_c, s_e) + \alpha_2 * NMCLCS_1(s_c, s_e) + \\ &\alpha_3 * NMCLCN_n(s_c, s_e) + \alpha_4 * NMCLCS_z(s_c, s_e) \end{aligned} \quad (1)$$

  where:
  - The $NLCS$ between s_c and s_e takes into account the length of both the shorter and the longer strings for normalization.

$$NLCS(s_c, s_e) = \frac{2 * len(LCS(s_c, s_e))}{len(s_c) + len(s_e)} \quad (2)$$

  - $NMCLCS_1$, $NMCLCS_n$ and $NMCLCS_z$ are variations of $NMCLCS$ starting from the first character, from any character and ending at the last character, respectively.

$$NMCLCS_p(s_c, s_e) = \frac{2 * len(MCLCS_p(s_c, s_e))}{len(s_c) + len(s_e)} \quad (3)$$

  with the character position index $p \in \{1, n, z\}$.
  - $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are weights and $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$. In our experiments, we heuristically set $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.25$.
- Bigram frequency:
  Let $f_2(.)$ be the bigram frequency depending the bigram dictionary (we apply the smoothing techniques to avoiding zero counting problem). Let $s_{c_{i-1}}$ and $s_{c_{i+1}}$ be the context syllables preceding and following the correction candidate s_{c_i} in the OCR text.
  The bigram frequency of the correction candidate s_{c_i} is calculated regarding its left and right context bigrams, $s_{c_{i-1}}s_{c_i}$ and $s_{c_i}s_{c_{i+1}}$. Let C be the set of correction candidates of the erroneous token s_e. The bigram frequency is computed as below:

$$BiFreq(s_{c_i}) = \frac{f_2(s_{c_{i-1}}s_{c_i}) + f_2(s_{c_i}s_{c_{i+1}})}{max_{s'_{c_i} \in C}\{f_2(s_{c_{i-1}}s'_{c_i}) + f_2(s'_{c_i}s_{c_{i+1}})\}} \quad (4)$$

  where each s'_{c_i} is a correction candidate in the candidate set C.
- Trigram frequency:
  Let $f_3(.)$ be the trigram frequency depending on the trigram dictionary.
  Similarly, the trigram frequency of the correction candidate s_{c_i} is calculated regarding all the context trigrams containing s_{c_i}. The trigram frequency is given by:

$$TriFreq(s_{c_i}) = \frac{\sum_{n=1}^{3} f_3(trigram_n(s_{c_i}))}{max_{s'_{c_i} \in C}\{\sum_{n=1}^{3} f_3(trigram_n(s'_{c_i}))\}} \quad (5)$$

where each s'_{c_i} is a correction candidate in the candidate set C.

- Edit Probability:
  - Let $f_{edit}(.)$ be the observed frequency of character edits depending on the CET table. And let E_{s_c} be the set of character edits applied to the erroneous token s_e to yield the correction candidate s_c.
  The edit probability of the correction candidate s_c is the normalized probability determined as the frequency product of the character edits yielding s_c over the maximum frequency product of the character edits yielding each among all the correction candidates:

$$EditProb(s_c) = \frac{\prod_{e \in E_{s_c}} f_{edit}(e)}{max_{s'_c \in C}\{\prod_{e' \in E_{s'_c}} f_{edit}(e')\}} \quad (6)$$

  where each e is a character edit in the set E_{s_c} of the correction candidate s_c, and each e' is a character edit in the set $E_{s'_c}$ of the correction candidate s'_c.

In order to score the correction candidate s_c of the erroneous token s_e, we take a weighted sum of the feature scores of s_c, given by:

$$\begin{aligned} FinalScore(s_c) = {} & p_1 * Sim(s_c) + \\ & p_2 * BiFreq(s_c) + \\ & p_3 * TriFreq(s_c) + \\ & p_4 * EditProb(s_c) \end{aligned} \quad (7)$$

where:

- $Sim(s_c)$, $BiFreq(s_c)$, $TriFreq(s_c)$ and $EditProb(s_c)$ are the measuring scores of syllable similarity, bigram frequency, trigram frequency and edit probability of the correction candidate s_c, respectively.

- The weights p_1, p_2, p_3 and p_4 are heuristically chosen and satisfied that $p_1 + p_2 + p_3 + p_4 = 1$.

In the following experiments, we suggest the highest-scored candidates as the final corrections for the erroneous tokens.

3 EXPERIMENTS AND RESULTS

3.1 Dataset

One reason of only a few previous works conducted in Vietnamese handwriting recognition is the lack of a benchmark dataset to evaluate and compare different approaches. These previous works have analyzed and evaluated the recognition results on their own datasets, which cause bias to compare among them.

The competition on Vietnamese online handwritten text recognition (VOHTR2018) using HANDS-VNOnDB[3] database [13] has been organized to encourage the studies on Vietnamese handwritten text recognition and to provide a benchmark database to verify and compare the recognition results of different approaches among the participants. HANDS-VNOnDB (VNOnDB in short) stores handwritten text collected from 200 Vietnamese. VNOnDB contains 1,146 Vietnamese paragraphs of handwritten text comprised of 7,296 lines, more than 480,000 strokes and more than 380,000 characters. VNOnDB provides the ink data and ground truth for paragraph, line and word levels (VNOnDB-Paragraph, VNOnDB-Line and VNOnDB-Word datasets, respectively). The ground truth texts are derived from the VTB corpus. Table 3 shows the number of lines, strokes and characters in the VNOnDB-Line dataset for the training, validation and test sets.

In our experiments, we only use the VNOnDB-Line dataset to evaluate our model on the OCR output lines of the baseline AED model to reduce the training time. The OCR output lines produced from the training and validation sets are used in combination with the corresponding GTs to construct the CET table. The OCR output lines from the test set are used to evaluate our model performance.

Table 3: The VNOnDB-Line Statistics.

	Training set	Validation set	Test set
# lines	4,433	1,229	1,634
# strokes	284,642	86,079	110,013
# characters	298,212	83,806	112,769

3.2 Evaluation Metric

We evaluate the performance of our proposed model on OCR output texts of Vietnamese handwriting using the character error rate (CER) and word error rate (WER) metrics, which are commonly employed for measuring the transcription results in recognition systems.

Let S be the set of source lines (ground truths of input lines) given by the VNOnDB-Line dataset, and let R be the set of corresponding correction lines resulted from the OCR post-processing stage. The normalized edit distance (NED) is given as below:

$$NED(S_i, R_i) = \frac{100}{|S_i|} * LVDist(S_i, R_i) \quad (8)$$

where S_i is the i-th source line in S, R_i is the corresponding correction line of S_i, and $LVDist$ is the Levenshtein edit distance that measures the minimum number of character edits between two character strings. The edit distance is normalized by the length of the source line S_i to avoid any bias related to the lengths of input lines.

The average normalized edit distance (ANED) for the entire dataset is then computed by:

$$ANED(S, R) = \frac{\sum_{i=1}^{N} NED(S_i, R_i)}{N} \quad (9)$$

where N is the number of source lines in S.

For CER, the edit distance is computed on the character level. For WER, the edit distance is computed on the word level. CER and WER are the inverse performance metrics. In other words, a low value shows high performance, and vice versa.

3.3 Experimental Results

All the teams participating in the VOHTR2018 competition have employed different BLSTM network architectures which were trained with Connectionist Temporal Classification (CTC). The Google team

[3]Source: http://tc11.cvc.uab.es/datasets/HANDS-VNOnDB2018_1

applies the basic preprocessing to the ink with scaling, normalizing, resampling, and representing by Bezier curves. The preprocessed data is then passed through multiple BLSTM layers. The resulted output of BLSTM layers is post-processed with *n*-gram language models at character and word levels on their private corpus. For the next participant, IVTOV team, they employ line segmentation before preprocessing. The extracted online features are then used to train a recognition network of two BLSTM layers with 100 cells in each layer. In the post-processing step, they apply the dictionary constraints to the output sequence of the recognition neural network. Another online recognition system is MyScript. The MyScript system preprocesses the digital ink by normalizing with a Bezier approximation, and correcting the slope and slant. It consists of two recognizers, one is the feed-forward network for predicting characters from segmented candidates, and one is the BLSTM network for predicting output text without segmentation. For post-processing, it uses a syllable-based *n*-gram language model trained on a large corpus (35 million tokens) including VTB, additional corpora and lexica; while in our model, we only base on the VTB corpus that contains about 2 million tokens.

Table 4 shows CER and WER of our OCR post-processing model relied on the AED model with DenseNet encoder, and other participants on the VNONDB-Line test set. Our approach achieves 4.17% of CER and 9.82% of WER, which improves 0.5% of CER and 3.5% of WER of the baseline AED model. Our model performs better than Nguyen et al.'s system and the GoogleTask2 system on CER. It also outperforms Nguyen et al.'s system, the GoogleTask2, and IVTOVTask2 systems on WER.

Table 4: The performance of recognition systems on the VNONDB-Line test set.

System	Corpus	CER (%)	WER (%)
GoogleTask2	Other	6.86	19.00
IVTOVTask2	VTB	3.24	14.11
MyScriptTask2_1	VTB	1.02	2.02
MyScriptTask2_2	VTB+ Other	1.57	4.02
Nguyen et al. [13]	None	7.17	N/A
AED model with DenseNet encoder [11]	None	4.67	13.33
Our post-OCR model based on the baseline AED	VTB	4.17	9.82

Our OCR post-processing model does not rely on any specified parameters of the baseline AED model (OCR) as well as original offline images. Instead, our model leverages the linguistic features and the error characteristics of the OCR output texts. In our experiments, we set the *edit_distance* parameter to 2 in order to reduce the computation time. Another reason for this is that a higher portion of OCR errors are originated from one or two character edit operations.

Tables 5 and 6 show correct and incorrect candidate suggestions for both non-syllable errors and real-syllable errors. For incorrect candidate suggestions, one reason is that the CET table is lacking in the appropriate character edits for correcting some erroneous tokens. This is due to the fact that the training data does not contain all necessary correction edits that occur in the test data. Another reason is our *n*-gram dictionaries are not large enough to include all needed *n*-grams that exist in the test data. In future work, we will try to construct the *n*-gram dictionaries with more external corpora.

Table 5: Examples of correct candidate suggestions.

OCR errors	Ground Truth	Examples
Non-syllable	cởi	"cới" corrected to "cởi"
	chưa	"nhưa" corrected to "chưa"
	khủng	"chủy" corrected to "khủng"
Real-syllable	thực phẩm	"thực thẩm" corrected to "thực phẩm"
	bữa cơm	"bia cơm" corrected to "bữa cơm"

Table 6: Examples of incorrect candidate suggestions.

OCR errors	Ground Truth	Examples
Non-syllable	để	"đẩo" corrected to "đẩu"
	Trên	"Cuên" corrected to "Cung"
	HIV	"HCS" corrected to "NCS"
Real-syllable	Tám On	"Tám on" corrected to "tám con"
	Trung tâm	"Trung tàm" corrected to "Trung làm"

4 CONCLUSION

We introduce an automatic hybrid approach for correcting both non-syllable and real-syllable errors in OCR output texts of unconstrained Vietnamese handwriting. The approach makes use of the characteristics of both Vietnamese language and OCR errors of handwritten texts. It is shown that our OCR post-processing model helps improve CER and WER of the baseline AED model by 0.5% and 3.5% respectively. Our model can be employed not only for OCR texts of Vietnamese handwriting, but also for any Vietnamese OCR texts regardless of text recognition systems.

However, the model still has lower results than the best performer (1.02% of CER and 2.02% of WER achieved by MyScript team) in the VOHTR2018 competition. In future research, we would like to further exploit the linguistic features with word *n*-gram contexts as well as to extend the correction candidate space with different methods. The new model will be evaluated on the VNOnDB-Line

and VNOnDB-Paragraph datasets to analyze its improvement on the CER and WER metrics.

REFERENCES

[1] Abdelhak Boukharouba and Abdelhak Bennia. 2017. Novel feature extraction technique for the recognition of handwritten digits. *Applied Computing and Informatics* 13, 1 (2017), 19–26. https://doi.org/10.1016/j.aci.2015.05.001

[2] Amit Choudhary, Rahul Rishi, and Savita Ahlawat. 2013. Off-line Handwritten Character Recognition Using Features Extracted from Binarization Technique. In *AASRI Procedia*, Vol. 4. 306–312. https://doi.org/10.1016/j.aasri.2013.10.045

[3] Apurva A. Desai. 2010. Gujarati Handwritten Numeral Optical Character Reorganization Through Neural Network. *Pattern Recognition* 43, 7 (2010), 2582–2589. https://doi.org/10.1016/j.patcog.2010.01.008

[4] Alex Graves, Marcus Liwicki, Horst Bunke, Jürgen Schmidhuber, and Santiago Fernández. 2008. Unconstrained On-line Handwriting Recognition with Recurrent Neural Networks. In *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (Eds.). Curran Associates, Inc., 577–584.

[5] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. 2009. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 5 (2009), 855–868. https://doi.org/10.1109/TPAMI.2008.137

[6] Aminul Islam and Diana Inkpen. 2008. Semantic Text Similarity Using Corpus-based Word Similarity and String Similarity. *ACM Trans. Knowl. Discov. Data* 2, 2, Article 10 (July 2008), 25 pages. https://doi.org/10.1145/1376815.1376819

[7] Aminul Islam and Diana Inkpen. 2009. Real-word Spelling Correction Using Google Web 1T n-gram Data Set. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*. ACM, New York, NY, USA, 1689–1692. https://doi.org/10.1145/1645953.1646205

[8] J. Jo, J. Lee, and Y. Lee. 2009. Stroke-Based Online Hangul/Korean Character Recognition. In *2009 Chinese Conference on Pattern Recognition*. 1–5. https://doi.org/10.1109/CCPR.2009.5343953

[9] I. Kissos and N. Dershowitz. 2016. OCR Error Correction Using Character Correction and Feature-Based Word Classification. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. 198–203. https://doi.org/10.1109/DAS.2016.44

[10] Anh Le Duc, Hung Nguyen, and Masaki Nakagawa. 2018. Recognizing Unconstrained Vietnamese Handwriting By Attention Based Encoder Decoder Model. In *2018 International Conference on Advanced Computing and Applications (ACOMP)*. 83–87. https://doi.org/10.1109/ACOMP.2018.00021

[11] Anh Le Duc, Hung Nguyen, and Masaki Nakagawa. 2020. An End-to-End Recognition System for Unconstrained Vietnamese Handwriting. *SN Computer Science* 1, 7 (Jan 2020). https://doi.org/10.1007/s42979-019-0001-4

[12] Jie Mei, Aminul Islam, Abidalrahman Moh'd, Yajing Wu, and Evangelos Milios. 2018. Statistical learning for OCR error correction. *Information Processing and Management* 54, 6 (2018), 874–887. https://doi.org/10.1016/j.ipm.2018.06.001

[13] Hung Tuan Nguyen, Cuong Tuan Nguyen, Pham The Bao, and Masaki Nakagawa. 2018. A database of unconstrained Vietnamese online handwriting and recognition experiments by recurrent neural networks. *Pattern Recognition* 78 (2018), 291–306. https://doi.org/10.1016/j.patcog.2018.01.013

[14] Hung Tuan Nguyen, Cuong Tuan Nguyen, and Masaki Nakagawa. 2018. ICFHR 2018 - Competition on Vietnamese Online Handwritten Text Recognition using HANDS-VNOnDB (VOHTR2018). In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 494–499. https://doi.org/10.1109/ICFHR-2018.2018.00092

[15] Phuong-Thai Nguyen, Xuan-Luong Vu, Thi-Minh-Huyen Nguyen, Van-Hiep Nguyen, and Hong-Phuong Le. 2009. Building a Large Syntactically-annotated Corpus of Vietnamese. In *Proceedings of the Third Linguistic Annotation Workshop (ACL-IJCNLP '09)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 182–185.

[16] T. T. H. Nguyen, M. Coustaty, A. Doucet, A. Jatowt, and N. V. Nguyen. 2018. Adaptive Edit-Distance and Regression Approach for Post-OCR Text Correction. *Dobreva M., Hinze A., Žumer M. (eds) Maturity and Innovation in Digital Libraries. ICADL 2018. Lecture Notes in Computer Science* 11279 (2018), 278–289. https://doi.org/10.1007/978-3-030-04257-8_29

[17] Duy Nguyen K. and The Bui D. 2008. On the problem of classifying Vietnamese online handwritten characters. In *2008 10th International Conference on Control, Automation, Robotics and Vision*. 803–808. https://doi.org/10.1109/ICARCV.2008.4795620

[18] Duy Nguyen K. and The Bui D. 2008. Recognizing Vietnamese Online Handwritten Separated Characters. In *2008 International Conference on Advanced Language Processing and Web Information Technology*. 279–284. https://doi.org/10.1109/ALPIT.2008.58

[19] P. A. Phuong, N. Q. Tao, and L. C. Mai. 2008. An Efficient Model for Isolated Vietnamese Handwritten Recognition. In *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. 358–361. https://doi.org/10.1109/IIH-MSP.2008.67

[20] L. Sun, T. Su, C. Liu, and R. Wang. 2016. Deep LSTM Networks for Online Chinese Handwriting Recognition. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 271–276. https://doi.org/10.1109/ICFHR.2016.0059

[21] De Cao Tran. 2012. An Efficient Method for On-line Vietnamese Handwritten Character Recognition. In *Proceedings of the Third Symposium on Information and Communication Technology (SoICT '12)*. ACM, New York, NY, USA, 135–141. https://doi.org/10.1145/2350716.2350737

[22] K. Sonu Varghese, Ajay James, and Saravanan Chandran. 2016. A Novel Tri-Stage Recognition Scheme for Handwritten Malayalam Character Recognition. *Procedia Technology* 24 (2016), 1333–1340. https://doi.org/10.1016/j.protcy.2016.05.137

[23] Cong Duy Vu Hoang and Ai Ti Aw. 2012. An Unsupervised and Data-driven Approach for Spell Checking in Vietnamese OCR-scanned Texts. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data (HYBRID '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 36–44.